

# Osnovi programiranja

*Beleške sa vežbi*

Smer *Računarstvo i informatika*  
Matematički fakultet, Beograd

Jelena Tomašević

December 25, 2005



# Sadržaj

<b>1</b>		<b>5</b>
1.1	Obrada teksta sa ulaza - putchar(), getchar() . . . . .	5
1.2	Formiranje HTML dokumenta . . . . .	6
1.3	Životni vek i oblast važenja promenjivih . . . . .	7
1.4	Konstante . . . . .	9
1.5	Konverzija . . . . .	11
1.5.1	Automatska konverzija . . . . .	11
1.5.2	EksPLICITNA konverzija . . . . .	11
1.5.3	Funkcije koje vrše konverziju . . . . .	12



# 1

1

## 1.1 Obrada teksta sa ulaza - putchar(), getchar()

**Primer 1** *Brojanje reči, linija i karaktera sa ulaza*

```
#include <stdio.h>

#define IN 1 /* inside a word */
#define OUT 0 /* outside a word */

/* count lines, words, and characters in input */

main()
{
    int c, nl, nw, nc, state;

    state = OUT;

    /*Postavljaju se sve tri promenljive na nulu*/
    /*Isto kao da smo napisali
    nl = (nw = (nc = 0));*/

    nl = nw = nc = 0;
    while ((c = getchar()) != EOF) {
        ++nc;
        if (c == '\n')
            ++nl;
        /*Operator || znaci OR*/
        if (c == ' ' || c == '\n' || c == '\t')
            state = OUT;
        else if (state == OUT) {
            state = IN;
            ++nw;
        }
    }
    printf("%d %d %d\n", nl, nw, nc);
}
```

---

<sup>1</sup>Zasnovano na primerima sa sajtova <http://www.matf.bg.ac.yu/~filip>, <http://www.matf.bg.ac.yu/~milena>

**Primer 2** Program prepisuje ulaz na izlaz pri čemu više blanko znakova zamenjuje jednim.

```
#include <stdio.h>

main()
{
    int znak; /*tekuci znak sa ulaza*/
    int preth; /*znak koji prethodi tekucem */
    preth='a'; /* inicijalizujemo vrednost prethodnog
               da bi prvi prolazak kroz petlju bio ispravan*/
    while ( (znak=getchar() ) !=EOF)
    {
        if (znak !=' ' || preth != ' ') putchar(znak);
        preth=znak;
    }
}
```

## 1.2 Formiranje HTML dokumenta

**Primer 3** Prilikm pokretanja programa koristiti redirekciju:

```
a.out >primer.html
```

kako bi se rezultat rada programa upisao u datoteku primer.html.

```
/*Ovaj program formira html dokument*/
#include <stdio.h>
main()
{
    printf("<html><head><title>Ova stranica
           je napravljena u c-u</title></head>");
    printf("<body><h3 align=center>
           Rezultat </h3></body></html>");
}
```

**Primer 4** Napisati program koji generiše html dokument sa engleskim alfabetom.

```
#include <stdio.h>
main()
{
    int i;
    printf("<HTML><head><title>Engleski alfabet</title><head>\n");
    printf("<body><ul>");
    for(i=0;i<=25;i++)
        printf("<li> %c %c \n",'A'+i,'a'+i);
    printf("</ul></body></HTML>\n"); }
```

**Primer 5** Napisati program koji generise html dokument koji prikazuje tablicu mnozenja za brojeve od 1 do 10.

```
#include<stdio.h>
main()
{
    int i,j;
    printf("<html><head><title>Mnozenje</title></head>");
```

```
printf("<body><h3 align=center> Rezultat </h3>");
printf("<table border=1>\n");

/* Prva vrsta sadrzi brojeve od 1 do 10*/
printf("<tr>");
printf("<th></th>");
for(i=1; i<=10; i++)
    printf("<th> %d </th>\n", i);
printf("</tr>");

for(i=1; i<=10; i++)
{
    printf("<tr>");

    /* Na pocetku svake vrste stampamo broj
    odgovarajuće vrste*/
    printf("<th>%d</th>", i);

    for(j=1; j<=10; j++)
        printf("<td>%d\t</td>\n", i*j);

    printf("</tr>");
}
printf("</table>");
printf("</body></html>");
return 0;
}
```

### 1.3 Životni vek i oblast važenja promenljivih

**Primer 6** *Demonstracija životnog veka i oblasti važenja promenljivih (scope).*

```
#include <stdio.h>

/* Globalna promenjiva */
int a = 0;

/* Uvecava se globalna promenjiva a */
void increase()
{
    a++;
    printf("increase::a = %d\n", a);
}

/* Umanjuje se lokalna promenjiva a. Globalna promenjiva zadržava svoju vrednost. */
void decrease()
{
    /* Ovo a je nezavisna promenjiva u odnosu na globalno a */
    int a = 0;
    a--;
    printf("decrease::a = %d\n", a);
}
```

```
}

void nonstatic_var()
{
    /* Nestaticke promenjive ne cuvaju vrednosti kroz pozive funkcije */
    int s=0;
    s++;
    printf("nonstatic::s=%d\n",s);
}

void static_var()
{
    /* Staticke promenjive cuvaju vrednosti kroz pozive funkcije.
       Inicijalizacija se odvija samo u okviru prvog poziva. */
    static int s=0;
    s++;
    printf("static::s=%d\n",s);
}

main()
{
    /* Promenjive lokalne za funkciju main */
    int i;
    int x = 3;

    printf("main::x = %d\n", x);

    for (i = 0; i<3; i++)
    {
        /* Promenjiva u okviru bloka je nezavisna od spoljne promenjive.
           Ovde se koristi promenjiva x lokalna za blok petlje koja ima
           vrednost 5, dok originalno x i dalje ima vrednost 3*/
        int x = 5;
        printf("for::x = %d\n", x);
    }

    /* U ovom bloku x ima vrednost 3 */
    printf("main::x = %d\n", x);

    increase();
    decrease();

    /* Globalna promenjiva a */
    printf("main::a = %d\n", a);

    /* Demonstracija nestatickih promenjivih */
    for (i = 0; i<3; i++)
        nonstatic_var();

    /* Demonstracija statickih promenjivih */
    for (i = 0; i<3; i++)
```



```

        static_var();
    }

```

Izlaz iz programa:

```

main::x = 3
for::x = 5
for::x = 5
for::x = 5
main::x = 3
increase::a = 1
decrease::a = -1
main::a = 1
nonstatic::s=1
nonstatic::s=1
nonstatic::s=1
static::s=1
static::s=2
static::s=3

```

**Primer 7** *Ilustruje vidljivost imena*

```

#include <stdio.h>

int i=10;

void main() {
    {
        int i=3;
        {
            int i=1;
            printf("%d\n", i);
        }
        printf("%d\n",i);
    }
    printf("%d\n",i);
}

```

## 1.4 Konstante

Koji su tipovi konstanti?

Celobrojna konstanta 1234 je tipa int.

Da bi konstanta bila long navodi se iza nje slovo L ili l, npr 123456789L.

Ako želimo da nam je konstanta unsigned onda na kraju pišemo U ili u.

Može i 1234567ul.

**Konstante realnih brojeva** sadrže decimalnu tačku(123.4) ili eksponent(1e-2) ili i jedno i drugo. Njihov tip je double osim ako nemaj sufiks f ili F kada je u pitanju float. L ili l označavaju long double.

**Oktalna konstanta** počinje sa 0, a heksadecimalna sa 0x. Npr broj 31 ili 037 - oktalno ili 0x1f - heksadecimalno. I one mogu da imaju U i L na kraju.

**Znakovna konstanta** je celobrojna vrednost napisana između jednostrukih navodnika. Vrednost date konstante je numerička vrednost datog znaka u računarskom setu znakova. Npr možemo da pišemo '0' umesto 48.

!!!Razlikovati znakovne konstante i niske koje se navode između dvostrukih navodnika!

Posebni znaci su znak za kraj reda, tab i slično.

Znakovna konstanta '\0' predstavlja znak čija je vrednost nula, treba ga razlikovati od '0' koja je znak čija je vrednost 48.

**Konstantna niska:** "Ja sam niska"

ili

"" /\*prazna niska\*/

Navodnici nisu deo niske već se koriste da bi je ograničili. Ako ih želimo unutar niske, oni se navode sa \".

Konstantna niska je polje znakova. Da bi se znalo gde je kraj niske, fizičko memorisanje liste zahteva da postoji jedan znak više koji označava kraj, to je '\0'. Da bi se odredila dužina niske mora se proći kroz celu nisku.

!!!Važno:

Koja je razlika između "x" i 'x'?

**Primer 8** Program uvodi niske karaktera terminisane nulom.

```
#include <stdio.h>

main()
{
    /* Poslednji bajt niske karaktera se postavlja na '\0' tj. 0 */
    char s[] = {'a', 'b', 'c', '\0' };

    /* Kraci nacin da se postigne prethodno */
    char t[] = "abc";

    /* Ispis niske s karakter po karakter*/
    int i;
    for (i = 0; s[i] != '\0'; i++)
        putchar(s[i]);
    putchar('\n');

    /* Ispis niske s koristeći funkciju printf */
    printf("%s\n", s);

    /* Ispis niske t karakter po karakter*/
    for (i = 0; t[i] != '\0'; i++)
        putchar(t[i]);
    putchar('\n');

    /* Ispis niske t koristeći funkciju printf */
    printf("%s\n", t);
}

Izlaz:
abc
abc
abc
```

abc

**Primer 9** *Primer funkcije koja izračunava dužinu niske znakova.*

```
#include <stdio.h>

int strlen(char s[])
{
    int i=0;

    while (s[i] != '\0')
        ++i;
    return i;
}

int main()
{
    printf("Duzina ove niske
           je: %d \n",strlen("Duzina ove niske je:"));
    return 0;
}
```

Izlaz:

Duzina ove niske je: 20

## 1.5 Konverzija

### 1.5.1 Automatska konverzija

Ako je jedan od operanada različit vrši se konverzija, uvek u smeru manjeg ka većem tipu

Naredba dodele:

```
int i=5;
float f=2.3;
f=i; /* f ce imati vrednost 5.0*/
```

obrnuto:

```
int i=5;
float f=2.3;
i=f; /* i ce imati vrednost 2*/
```

### 1.5.2 Eksplicitna konverzija

(tip)<izraz>

```
float x;
x=2.3+4.2;          /* x ce imati vrednost 6.5 */
x=(int)2.3+(int)4.2; /* x ce imati vrednost 6 */
x=(int)2.3*4.5;     /* x ce imati vrednost 9.0 jer zbog prioriteta
                    operatora konverzije prvo ce biti izvršena
                    konverzija broja 2.3 u 2 pa tek onda izvršeno
```

```

                                mnozenje. */
x=(int)(2.3*4.5)    /* x ce imati vrednost 10.0 */

```

**Primer 10** *Kako izbeći celobrojno deljenje*

```

int a,b;
float c;
a = 5;
b = 2;
c = a/b; /* Celobrojno deljenje, c=2*/
c = (1.0*a)/b; /* Implicitna konverzija: 1.0*a je realan
                broj pa priliko deljenja sa b dobija se
                realan rezultat c=2.5*/
c = (0.0+a)/b; /* Implicitna konverzija: (0.0+a) je realan
                broj pa priliko deljenja sa b dobija se
                realan rezultat c=2.5*/
c = (float)a/(float)b; /* Eksplicitna konverzija*/

```

### 1.5.3 Funkcije koje vrše konverziju

**Primer 11**

```

#include <stdio.h>
main()
{
int vrednost;
vrednost='A';
printf("Veliko slovo\n karakter=%3c\nvrednost=%3d\n",vrednost,vrednost);
vrednost='a';
printf("Malo\n karakter=%3c\nvrednost=%3d\n",vrednost,vrednost);
}

```

Izlaz (u slucaju ASCII):

```

Veliko slovo
karakter= A
vrednost= 65
Malo
karakter= a
vrednost= 97

```

**Primer 12** *Funkcija koja konvertuje velika slova u mala slova.*

```

#include<stdio.h>

/* Konvertuje karakter iz velikog u malo slovo */
char lower(char c)
{
if (c >= 'A' && c <= 'Z')
    return c - 'A' + 'a' ;
else
    return c;
}

main()

```

```
{
char c;
printf("Unesi neko veliko slovo:\n");
scanf("%c", &c);
printf("Odgovarajuće malo slovo je %c\n", lower(c));
}
```

Izlaz:

```
Unesi neko veliko slovo:
J
Odgovarajuće malo slovo je j
```

**Primer 13** *Konvertovanje niske cifara u ceo broj.*

```
#include<stdio.h>

/* atoi: konvertuje s u ceo broj */
int atoi(char s[])
{
int i, n;
n = 0;
for (i = 0; (s[i] >= '0') && (s[i] <= '9'); ++i)
    n = 10 * n + (s[i] - '0');
return n;
}

main()
{
int n;
n = atoi("234");
printf("\nN je : %d\n",n);
}
```

Izlaz:

```
N je : 234
```

**Primer 14** *btoi - konverzija iz datog brojnog sistema u dekadni.*

```
#include <stdio.h>
#include <ctype.h>

/* Pomocna funkcija koja izracunava vrednost koju predstavlja karakter u datoj osnovi
   Funkcija vraca -1 ukoliko cifra nije validna.

   Npr.
   cifra 'B' u osnovi 16 ima vrednost 11
   cifra '8' nije validna u osnovi 6

*/

int digit_value(char c, int base)
```

```

{
    /* Proveravamo obicne cifre */
    if (isdigit(c) && c < '0'+base)
        return c-'0';

    /* Proveravamo slovne cifre za mala slova */
    if ('a'<=c && c < 'a'+base-10)
        return c-'a'+10;

    /* Proveravamo slovne cifre za velika slova */
    if ('A'<=c && c < 'A'+base-10)
        return c-'A'+10;

    return -1;
}

/* Funkcija izracunava vrednost celog broja koji je zapisan u datom
nizu karaktera u datoj osnovi. Za izracunavanje se koristi Hornerova shema.
*/
int btoi(char s[], int base)
{
    int sum = 0;

    /* Obradjuju se karakteri sve dok su cifre */
    int i, vr;
    for (i = 0; (vr = digit_value(s[i], base)) != -1; i++)
        sum = base*sum + vr;

    return sum;
}

main()
{
    char bin[] = "11110000";
    char hex[] = "FF";

    printf("Dekadna vrednost binarnog broja %s je %d\n", bin, btoi(bin, 2));
    printf("Dekadna vrednost heksadekadnog broja %s je %d\n", hex, btoi(hex, 16));
}

```

Izlaz:

Dekadna vrednost binarnog broja 11110000 je 240

Dekadna vrednost heksadekadnog broja FF je 255

**Primer 15** Program vrši konverziju iz dekadnog brojnog sistema u datu osnovu

```

#include<stdio.h>
#define OSNOVA 16
main()
{
    int x; /* Broj cija se konverzija vrši */

```

```

int ostaci[32]; /* Niz ostataka pri deljenju sa osnovom */
int i = 0;

    /* Unosi se dekadni broj */
scanf("%d",&x);

/* Srz algoritma konverzije */
while(x>0)
{
    /* novi ostatak se dodaje u pomocni niz */
    ostaci[i++] = x%OSNOVA;
    x/=OSNOVA;
}

/* Niz se ispisuje unatrag */
for (i--; i>=0; i--)
    if (ostaci[i]<=10) /* Slucaj kada je cifra dekadna */
        printf("%d",ostaci[i]);
    else /* Slucaj kada cifra prevazilazi dekadni opseg */
        printf("%c",'A'+ostaci[i]-10);

printf("\n");
}

```

### Zadaci za vežbu

**Zadatak 1** 1. Napisati funkciju koja računa  $k$ -ti stepen prirodnog broja  $n$ .

2. Napisati program koji formira HTML dokument koji sadrži tabelu u čijoj se  $j$ -toj ćeliji  $i$ -te kolone nalazi vrednost  $i^j$ .

3. Napisati program koji za uneti niz koeficijenata  $a[i]$  i uneti broj  $x$  računa vrednost polinoma  $a_n * x^n + a_{n-1} * x^{n-1} + \dots + a_1 * x + a_0$

**Zadatak 2** Napisati program koji formira HTML dokument koji sadrži numerisanu lista čija je svaka stavka malo slovo engleske abecede i njemu odgovarajući kod.

**Zadatak 3** Napisati program koji broji pojavljivanja samoglasnika i suglasnika u tekstu koji se unosi sa standardnog ulaza (koristiti naredbu `switch`).

**Zadatak 4** Napisati program koji broji pojavljivanja za svako od slova engleske abecede u tekstu koji se unosi sa standardnog ulaza i štampa rezultat na standardni izlaz.

**Zadatak 5** 1. Napisati funkciju koja konvertuje malo u veliko slovo.

2. Napisati program koji prepisuje ulaz na izlaz pri čemu se sva mala slova konvertuju u velika.